In [598…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [599…

```
data = pd.read_csv("Downloads/Tuesday.csv")
```

In [600…

```
data.set_index("Unnamed: 0", inplace=True)
```

In [601…

```
data.index.rename("id", inplace=True)
```

In [602…

```
data
```

Out[602…

	class	x	y	w	h	frameNo
id						
4068	5	0.876389	0.517969	0.241667	0.135937	4514
3655	5	0.870139	0.517187	0.254167	0.132812	4515
4211	5	0.861111	0.516016	0.266667	0.135156	4516
5745	5	0.845139	0.514453	0.298611	0.135156	4518
5320	5	0.838194	0.512891	0.315278	0.135156	4519
...	...	...	...	...	...	...
1347	5	0.815972	0.492188	0.368056	0.117188	500661
1978	5	0.823611	0.491797	0.344444	0.116406	500662
2350	5	0.831250	0.492188	0.331944	0.115625	500663
144	5	0.864583	0.496094	0.270833	0.117188	500667
769	5	0.893750	0.494531	0.212500	0.115625	500670

9417 rows × 6 columns

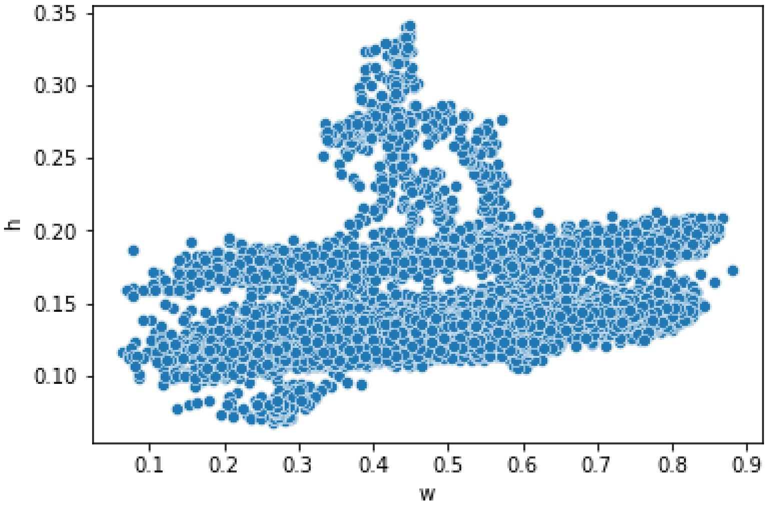
In [603…

```
sb.scatterplot(data["w"],data["h"])
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[603…

```
<AxesSubplot:xlabel='w', ylabel='h'>
```



In [604…

```
from sklearn.mixture import GaussianMixture
```

In [605…

```
gm = GaussianMixture(n_components=3, random_state=0, covariance_type="full", init_pa
```

In [606…

```
dat = data[["w","h"]]
```

In [607…

```
gm.fit(dat)
```

Out[607…

```
GaussianMixture(init_params='random', n_components=3, random_state=0)
```

In [608…

```
data["cluster"] = gm.predict(data[["w","h"]])
```

In [609…

```
data["cluster"].value_counts()
```

Out[609…

```
2    6719
1    1999
0     699
Name: cluster, dtype: int64
```

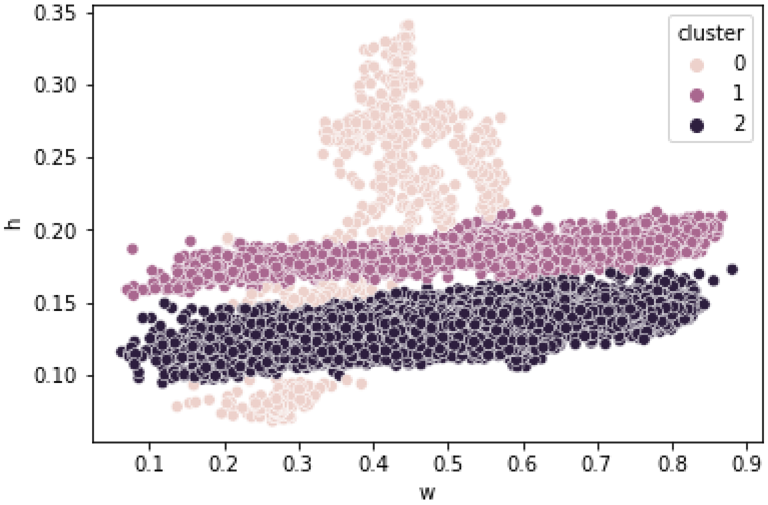
In [610…

```
sb.scatterplot(data["w"],data["h"], hue=data["cluster"])
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[610…

```
<AxesSubplot:xlabel='w', ylabel='h'>
```



Cluster 2 represents the similar width and height of "red" and "blue" buses at NTU

Cluster 1 repesents the "199" width and height

For Cluster 0, three types of observations were noted: 1) Construction minijeeps with height<0.10 frame units which were missclassified as buses

2) Buses reversing and coming near the camera, which gave these instances frame height>0.20 frame units

3) Single red and blue bus frames, with a "boxing" issue, resulting in erraneous frame width and height

We are only going to consider Red Bus data for this project

In [611…

```
busdata = data[data["cluster"]==2]
busdata.drop(columns={"cluster"},inplace=True)
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:43
08: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  return super().drop(
```

In [612…

```
busdata.drop(columns={"class"}, inplace=True)
```

In [613…

```
busdata.head(n=15)
```

Out[613…

	x	y	w	h	frameNo
id					
4068	0.876389	0.517969	0.241667	0.135937	4514
3655	0.870139	0.517187	0.254167	0.132812	4515
4211	0.861111	0.516016	0.266667	0.135156	4516
5745	0.845139	0.514453	0.298611	0.135156	4518
5320	0.838194	0.512891	0.315278	0.135156	4519

	x	y	w	h	frameNo
id					
5887	0.829167	0.512500	0.330556	0.135937	4520
6287	0.820833	0.512891	0.347222	0.136719	4521
6844	0.812500	0.512109	0.366667	0.138281	4522
6424	0.802083	0.512500	0.381944	0.140625	4523
5761	0.796528	0.511328	0.395833	0.139844	4524
5301	0.788194	0.510156	0.418056	0.139062	4525
4746	0.782639	0.509375	0.434722	0.137500	4526
5203	0.777083	0.508594	0.445833	0.137500	4527
4096	0.768056	0.508203	0.463889	0.138281	4528
3627	0.759722	0.507422	0.477778	0.138281	4529

In [614…

```
busdata["w"].mean()
```

Out[614…

```
0.4939816210150339
```

In [615…

```
busdata["w"].var()
```

Out[615…

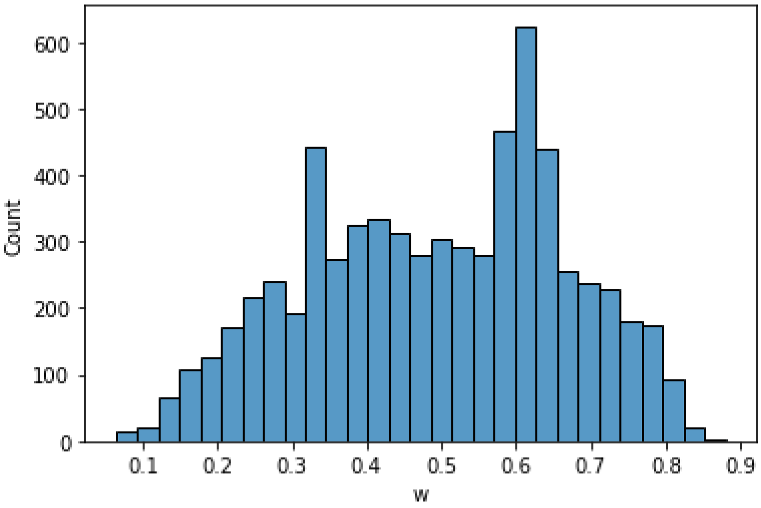
```
0.029316436083795697
```

In [616…

```
sb.histplot(busdata["w"])
```

Out[616…

```
<AxesSubplot:xlabel='w', ylabel='Count'>
```

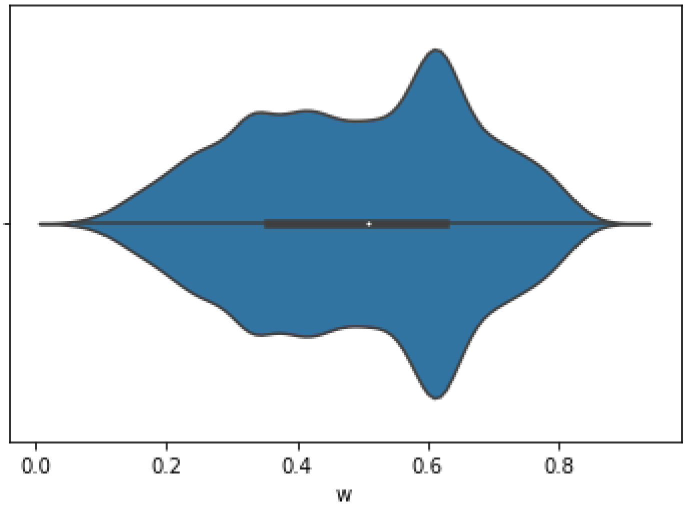


In [617…

```
sb.violinplot(busdata["w"])
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.1
2, the only valid positional argument will be `data`, and passing other arguments wi
thout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel='w'>
```

Out[617…



In [618…

```
busdata["h"].mean()
```

Out[618…

```
0.13303052299449306
```

In [619…

```
busdata["h"].var()
```

Out[619…

```
0.00015682376278292637
```

Ooops! Seems like we didn't completely remove that cluster of minijeeps, making our distribution of width bimodal

Going over all procedures again, with clusters = 4 this time;

In [620…

```
data = pd.read_csv("Downloads/Tuesday.csv")
data.set_index("Unnamed: 0", inplace=True)
data.index.rename("id", inplace=True)
gm = GaussianMixture(n_components=4, random_state=0, covariance_type="full")
dat = data[["w","h"]]
gm.fit(dat)
data["cluster"] = gm.predict(data[["w","h"]])
data["cluster"].value_counts()
```

Out[620…

```
3    5721
1    2001
0    1243
2     452
Name: cluster, dtype: int64
```

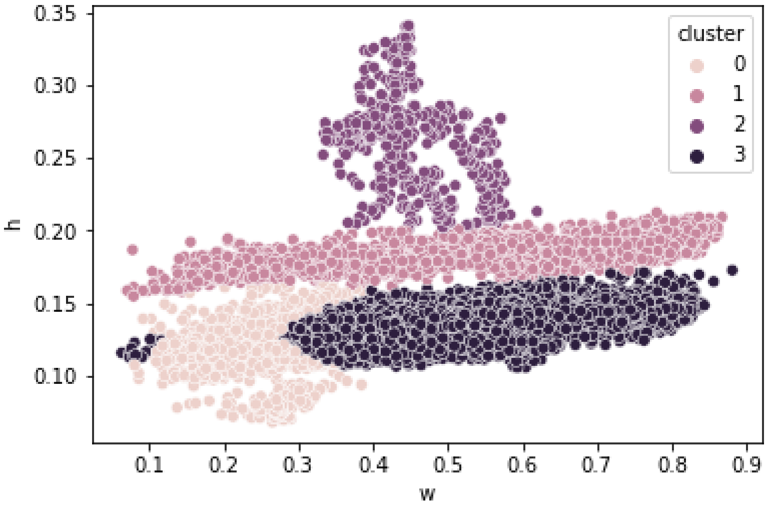
In [621…

```
sb.scatterplot(data["w"],data["h"], hue=data["cluster"])
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[621…

```
<AxesSubplot:xlabel='w', ylabel='h'>
```



In [622…

```
busdata = data[data["cluster"]==3]
busdata.drop(columns={"cluster"},inplace=True)
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/pandas/core/frame.py:43
08: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  return super().drop(
```

In [623…

```
busdata.drop(columns={"class"}, inplace=True)
```

In [624…

```
busdata.head(n=10)
```

Out[624…

	x	y	w	h	frameNo
id					
5745	0.845139	0.514453	0.298611	0.135156	4518
5320	0.838194	0.512891	0.315278	0.135156	4519
5887	0.829167	0.512500	0.330556	0.135937	4520
6287	0.820833	0.512891	0.347222	0.136719	4521
6844	0.812500	0.512109	0.366667	0.138281	4522
6424	0.802083	0.512500	0.381944	0.140625	4523
5761	0.796528	0.511328	0.395833	0.139844	4524
5301	0.788194	0.510156	0.418056	0.139062	4525
4746	0.782639	0.509375	0.434722	0.137500	4526
5203	0.777083	0.508594	0.445833	0.137500	4527

In [625…

```
busdata["w"].mean()
```

Out[625…

```
0.5403105548155933
```

In [626…

```
busdata["w"].var()
```

Out[626…

```
0.019499269369984213
```

That's it! The cluster variance for width has gone down

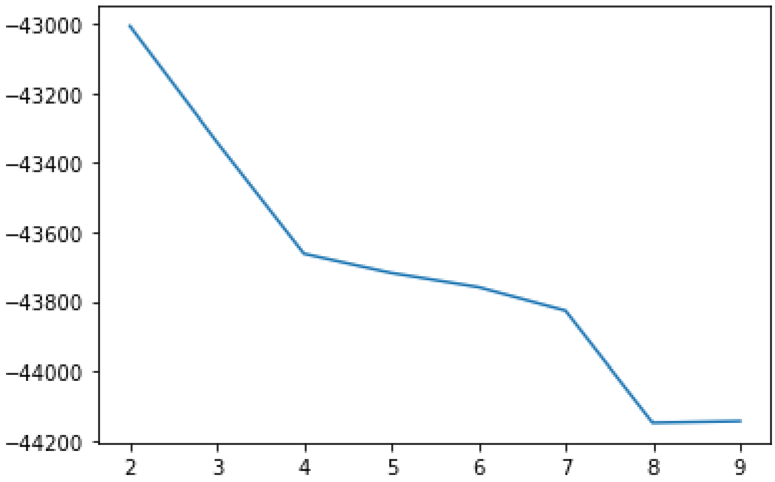
In [627…

```
aic = []
for i in range(2,10):
    gm = GaussianMixture(n_components=i, random_state=0, covariance_type="full")
    gm.fit(busdata[["w","h"]])
    aic.append(gm.aic(busdata[["w","h"]]))

index = []
for i in range(2,10):
    index.append(i)

sb.lineplot(index,aic)
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[627…

```
<AxesSubplot:>
```

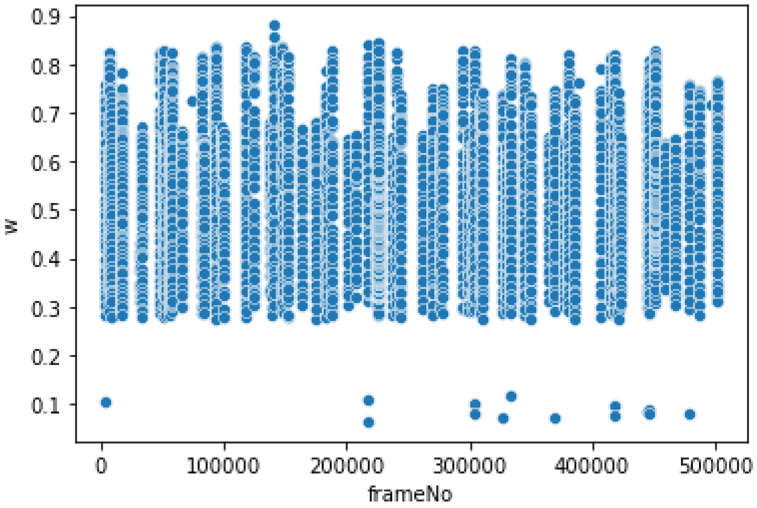


In [628…

```
sb.scatterplot(x=busdata["frameNo"],y=busdata["w"])
```

Out[628…

```
<AxesSubplot:xlabel='frameNo', ylabel='w'>
```



In [629…

```
aic = []
for i in range(2,301):
    gm = GaussianMixture(n_components=i, random_state=0, covariance_type="tied")
    gm.fit(busdata[["w","frameNo"]])
    aic.append(gm.aic(busdata[["w","frameNo"]]))
```

In [630…

```
index = []
for i in range(2,301):
    index.append(i)
```

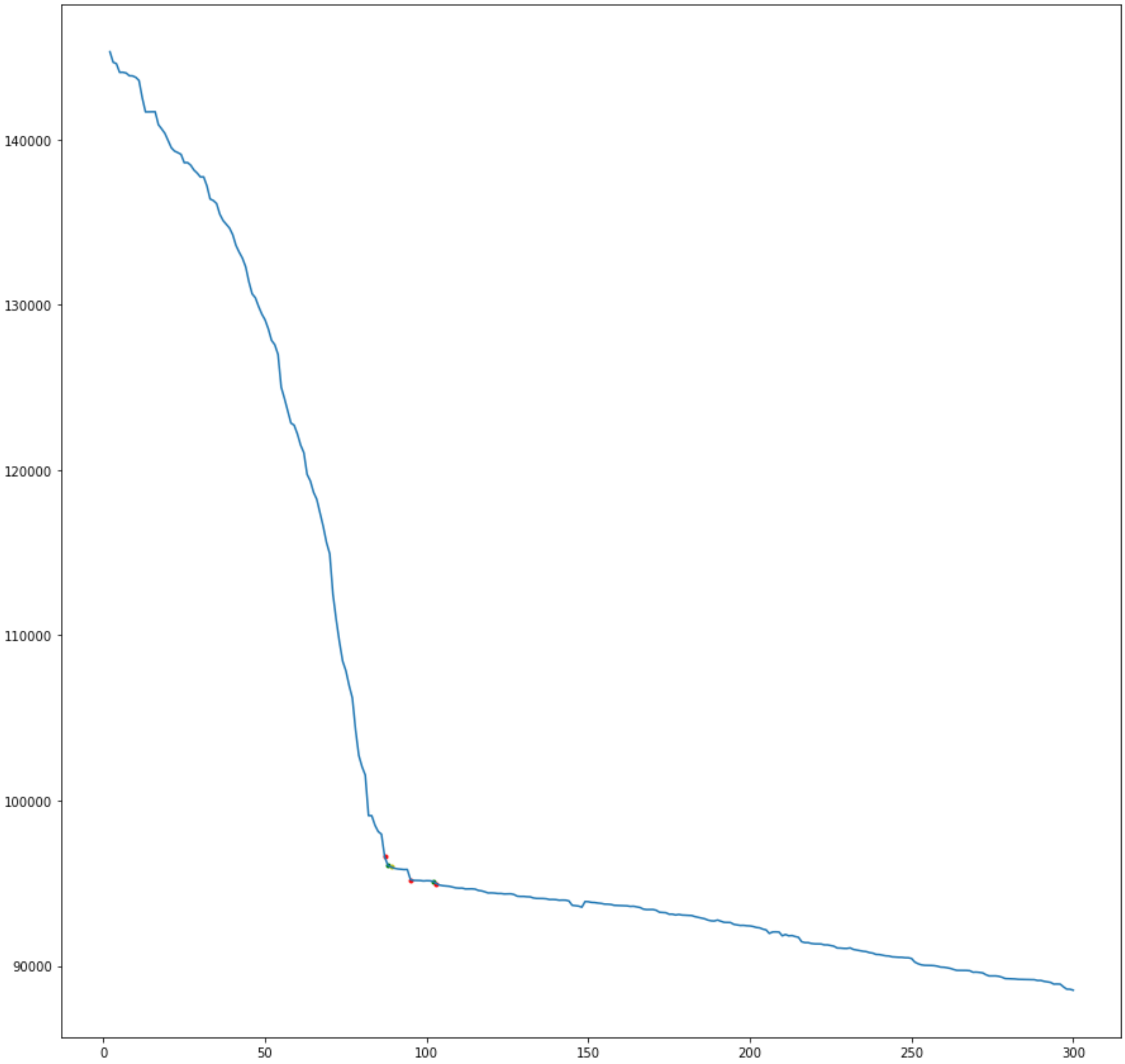
In [809…

```
plt.subplots(figsize=(15,15))
plt.plot(87,aic[85], 'r.')
plt.plot(88,aic[86], 'g.')
plt.plot(89,aic[87], 'y.')
plt.plot(95,aic[93], 'r.')
plt.plot(102,aic[100], 'g.')
plt.plot(103,aic[101], 'r.')
sb.lineplot(index,aic)
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[809…

```
<AxesSubplot:>
```



```
In [876…  aic[93]

Out[876…  95211.868436626
```

```
In [810…  aic[80:100]

Out[810…  [99091.58692608768,
           99087.5048097385,
           98510.29275722278,
           98125.25462841416,
           97967.86537607241,
           96601.92901573656,
           96072.53495059785,
           95992.01737829906,
           95903.55085564288,
           95867.74606860698,
           95843.59652254554,
           95826.2029075749,
           95832.36572873734,
           95211.868436626,
           95172.4958123339,
           95169.68348119213,
           95163.22222172245,
           95138.02085144876,
           95152.27523391298,
           95147.2994676051]
```

In [926…

```
gm = GaussianMixture(n_components=88, random_state=0, covariance_type="tied")
gm.fit(busdata[["w","frameNo"]])
busdata["cluster"]= gm.predict(busdata[["w","frameNo"]])
```

```
<ipython-input-926-2cc3bb4653db>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  busdata["cluster"]= gm.predict(busdata[["w","frameNo"]])
```

In [927…

```
sample = busdata[500:1000]
```

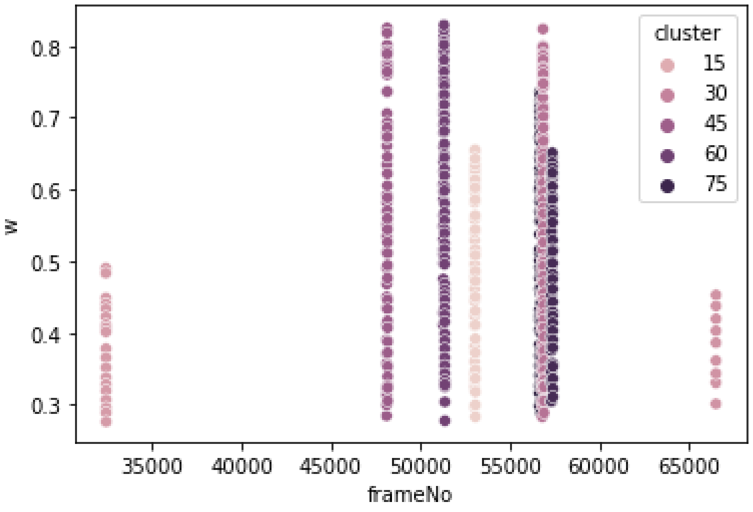
In [928…

```
sb.scatterplot(sample["frameNo"],sample["w"], hue=sample["cluster"])
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[928…

```
<AxesSubplot:xlabel='frameNo', ylabel='w'>
```



In [929…

```
busdata["frameNo"].value_counts()[:20]
```

Out[929…

```
500629    2
152694    2
500638    2
500630    2
500632    2
500643    2
500628    2
152696    2
500644    2
188094    2
500631    2
152692    2
57344     1
118075    1
7481      1
```

```
7485      1
118079    1
7489      1
459161    1
7477      1
Name: frameNo, dtype: int64
```

In [942…

```
500629/(30*60)
```

Out[942…

```
278.1272222222222
```

In [930…

```
busdata[busdata["frameNo"]==500629]
```

Out[930…

id	x	y	w	h	frameNo	cluster	type
4271	0.383333	0.487500	0.766667	0.129688	500629	13	red
4272	0.588889	0.484375	0.338889	0.110937	500629	13	blue

In [931…

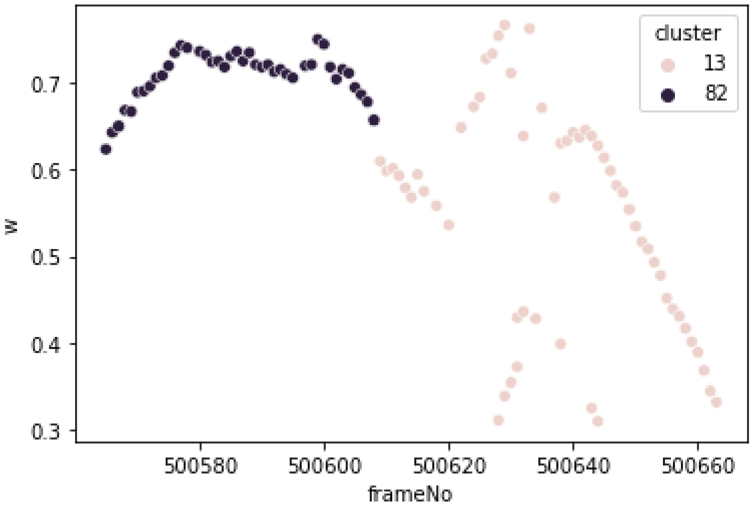
```
sample = busdata[-100:]
print(busdata["frameNo"])
sb.scatterplot(sample["frameNo"],sample["w"], hue=sample["cluster"])
```

```
id
5745      4518
5320      4519
5887      4520
6287      4521
6844      4522
         ...
1112    500659
1736    500660
1347    500661
1978    500662
2350    500663
Name: frameNo, Length: 5721, dtype: int64
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[931…

```
<AxesSubplot:xlabel='frameNo', ylabel='w'>
```



In [932…

```python
print(len(busdata["cluster"].value_counts()))
busdata["cluster"].value_counts()[60:100]
```

```
88
```

Out[932…

```
76    55
35    55
74    55
11    54
61    53
50    53
60    53
45    52
25    52
39    52
66    51
43    51
29    50
62    49
15    49
51    49
22    49
1     48
37    47
41    47
57    46
54    46
78    41
40    41
86     4
70     1
71     1
77     1
Name: cluster, dtype: int64
```

We conclude that we saw 88 - 4= 84 buses

Determination of Red vs Blue Bus

In [933…

```python
for i in range(1,len(busdata)):
    if(busdata.at[busdata.index[i],"cluster"]==busdata.at[busdata.index[i-1],"cluste
        if(busdata.at[busdata.index[i],"x"]>busdata.at[busdata.index[i-1],"x"]):
            busdata.at[busdata.index[i],"type"]="blue"
        else:
            busdata.at[busdata.index[i],"type"]="red"
    else:
        busdata.at[busdata.index[i],"type"]="error"

for i in range(1,len(busdata)):
    if(busdata.at[busdata.index[i],"type"]=="error"):
        busdata.at[busdata.index[i],"type"]=busdata.at[busdata.index[i+1],"type"]
```

In [934…

```
busdata[["cluster","type"]].value_counts()
```

Out[934…

```
cluster  type 
12       red      110
6        red      100
36       red       97
65       blue      97
75       red       94
                 ... 
71       error      1
70       error      1
68       blue       1
58       red        1
77       error      1
Length: 126, dtype: int64
```

In [935…

```
data_cluster = busdata.set_index(["cluster"])
```

In [936…

```
data_cluster.head(10)
```

Out[936…

	x	y	w	h	frameNo	type
cluster						
49	0.845139	0.514453	0.298611	0.135156	4518	red
49	0.838194	0.512891	0.315278	0.135156	4519	red
49	0.829167	0.512500	0.330556	0.135937	4520	red
49	0.820833	0.512891	0.347222	0.136719	4521	red
49	0.812500	0.512109	0.366667	0.138281	4522	red
49	0.802083	0.512500	0.381944	0.140625	4523	red
49	0.796528	0.511328	0.395833	0.139844	4524	red
49	0.788194	0.510156	0.418056	0.139062	4525	red
49	0.782639	0.509375	0.434722	0.137500	4526	red
49	0.777083	0.508594	0.445833	0.137500	4527	red

In [937…

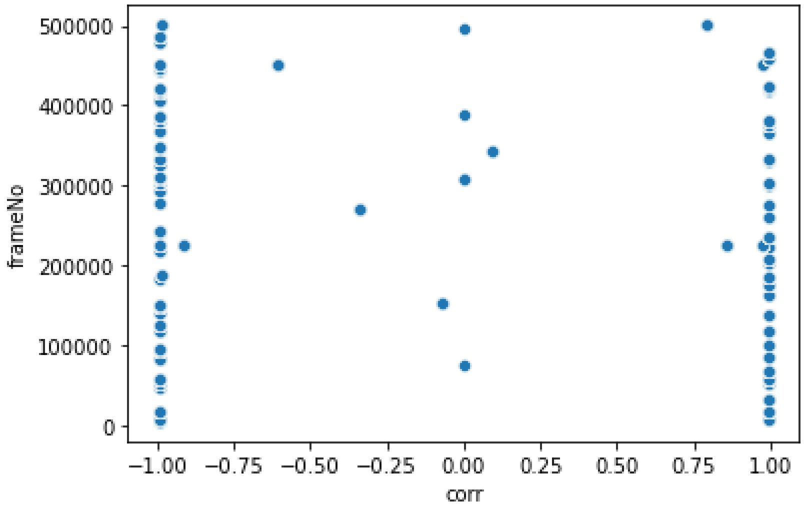
```
for i in pd.Series(data_cluster.index).unique():
    if(len(data_cluster.loc[i])>10):
        data_cluster.loc[i,"corr"]=data_cluster.loc[i,["x","frameNo"]].corr()["x"]["
    else:
        data_cluster.loc[i,"corr"]=0
```

In [938…

```
sb.scatterplot(x=data_cluster["corr"],y=data_cluster["frameNo"])
```

Out[938…

```
<AxesSubplot:xlabel='corr', ylabel='frameNo'>
```



In [939…

```
data_cluster[(data_cluster["corr"]<0.5) & (data_cluster["corr"]>-0.5)][50:100]
```

Out[939…

	x	y	w	h	frameNo	type	corr
cluster							
4	0.259722	0.486719	0.513889	0.150000	152683	red	-0.070827
4	0.259722	0.483594	0.511111	0.151562	152684	red	-0.070827
4	0.259028	0.480078	0.515278	0.153906	152685	red	-0.070827
4	0.252778	0.481641	0.494444	0.139844	152686	red	-0.070827
4	0.231944	0.485156	0.450000	0.148438	152687	red	-0.070827
4	0.222222	0.481641	0.444444	0.136719	152688	red	-0.070827
4	0.213889	0.482422	0.425000	0.142969	152689	red	-0.070827
4	0.200694	0.480469	0.401389	0.135937	152690	red	-0.070827
4	0.330556	0.480469	0.627778	0.157813	152691	blue	-0.070827
4	0.186111	0.480859	0.363889	0.132031	152692	red	-0.070827
4	0.487500	0.479297	0.341667	0.125781	152692	blue	-0.070827
4	0.355556	0.482031	0.672222	0.157813	152693	red	-0.070827
4	0.162500	0.483203	0.319444	0.130469	152694	red	-0.070827
4	0.341667	0.486328	0.683333	0.141406	152694	blue	-0.070827
4	0.369444	0.481641	0.669444	0.152344	152695	blue	-0.070827
4	0.423611	0.481641	0.605556	0.135156	152696	blue	-0.070827
4	0.145833	0.483203	0.283333	0.130469	152696	red	-0.070827
4	0.410417	0.487891	0.687500	0.153906	152697	blue	-0.070827
4	0.482639	0.490625	0.531944	0.143750	152698	blue	-0.070827
4	0.488194	0.491406	0.618056	0.139062	152700	blue	-0.070827
4	0.489583	0.483984	0.648611	0.149219	152701	blue	-0.070827
4	0.521528	0.491797	0.629167	0.142969	152702	blue	-0.070827
4	0.531944	0.494922	0.625000	0.142969	152703	blue	-0.070827

	x	y	w	h	frameNo	type	corr
cluster							
4	0.550000	0.493750	0.638889	0.145313	152704	blue	-0.070827
4	0.563194	0.490234	0.631944	0.133594	152705	blue	-0.070827
4	0.595833	0.487891	0.666667	0.139844	152706	blue	-0.070827
4	0.607639	0.494922	0.631944	0.138281	152707	blue	-0.070827
4	0.620833	0.491016	0.675000	0.153906	152708	blue	-0.070827
4	0.640278	0.493750	0.633333	0.132812	152709	blue	-0.070827
4	0.652778	0.489844	0.658333	0.148438	152710	blue	-0.070827
4	0.675694	0.492578	0.640278	0.130469	152711	blue	-0.070827
4	0.676389	0.492578	0.647222	0.142969	152712	blue	-0.070827
4	0.696528	0.493359	0.606944	0.136719	152713	blue	-0.070827
4	0.702083	0.495703	0.595833	0.144531	152714	blue	-0.070827
4	0.711806	0.498828	0.576389	0.136719	152715	blue	-0.070827
4	0.712500	0.495703	0.566667	0.149219	152716	blue	-0.070827
4	0.728472	0.499219	0.531944	0.134375	152717	blue	-0.070827
4	0.732639	0.498047	0.526389	0.152344	152718	blue	-0.070827
4	0.744444	0.501562	0.502778	0.139062	152719	blue	-0.070827
4	0.753472	0.498437	0.490278	0.134375	152720	blue	-0.070827
4	0.759028	0.500391	0.473611	0.144531	152721	blue	-0.070827
4	0.764583	0.497656	0.456944	0.126563	152722	blue	-0.070827
4	0.775694	0.501953	0.434722	0.147656	152723	blue	-0.070827
4	0.784722	0.504687	0.416667	0.137500	152724	blue	-0.070827
4	0.790972	0.506250	0.404167	0.137500	152725	blue	-0.070827
4	0.800000	0.502734	0.383333	0.144531	152726	blue	-0.070827
4	0.806944	0.496875	0.372222	0.126563	152727	blue	-0.070827
4	0.823611	0.496875	0.336111	0.125000	152729	blue	-0.070827
4	0.859722	0.498047	0.280556	0.128906	152732	blue	-0.070827
44	0.853472	0.516016	0.293056	0.125781	269630	red	-0.338385

In [940…

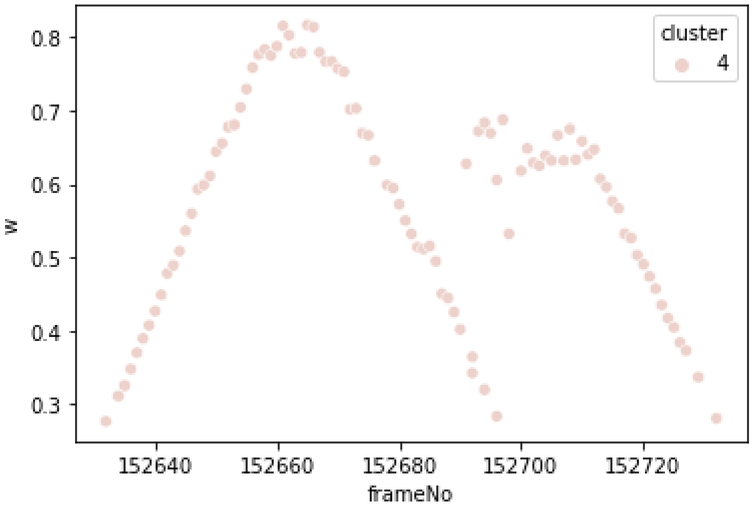
```
smaple = data_cluster[data_cluster.index==4]
```

In [941…

```
sb.scatterplot(smaple["frameNo"],smaple["w"], hue=smaple.index)
```

```
/Users/dhruvchopra/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel='frameNo', ylabel='w'>
```

Out[941…



In [925…

```
data_cluster.loc[i,["x","frameNo"]].corr()["x"]["frameNo"]
```

Out[925…

```
0.7891658067277243
```

In [ ]: